

SEI Webinar Series:

Agile Development & Software Architecture – Crossing the Great Divide

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Nanette Brown
4/22/2010



Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 22 APR 2010		2. REPORT TYPE		3. DATES COVERED 00-00-2010 to 00-00-2010	
4. TITLE AND SUBTITLE Agile Development & Software Architecture - Crossing the Great Divide				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Carnegie Mellon University ,Software Engineering Institute,Pittsburgh,PA,15213				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 60	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

Today's Presenter

Nanette Brown is a Visiting Scientist with the Software Engineering Institute's Research, Technology, and System Solutions Program and is a Principal Consultant with NoteWell Consulting. She is currently engaged in an SEI Research Project on "Communicating the Value of Architecting within Agile Development" as well as other activities focusing on architecture within an Agile context.

Previously, Nanette worked at Pitney Bowes Inc., most recently as Director of Architecture and Quality Management, where she was responsible for design and implementation of a customized SDLC that blended RUP and Agile practices. Nanette has presented at multiple industry conferences including SD Best Practices and Project World and the World Conference of Business Analysts on topics such as Facilitated Iteration Planning and the SEI scenario-based approach to specify quality attributes.

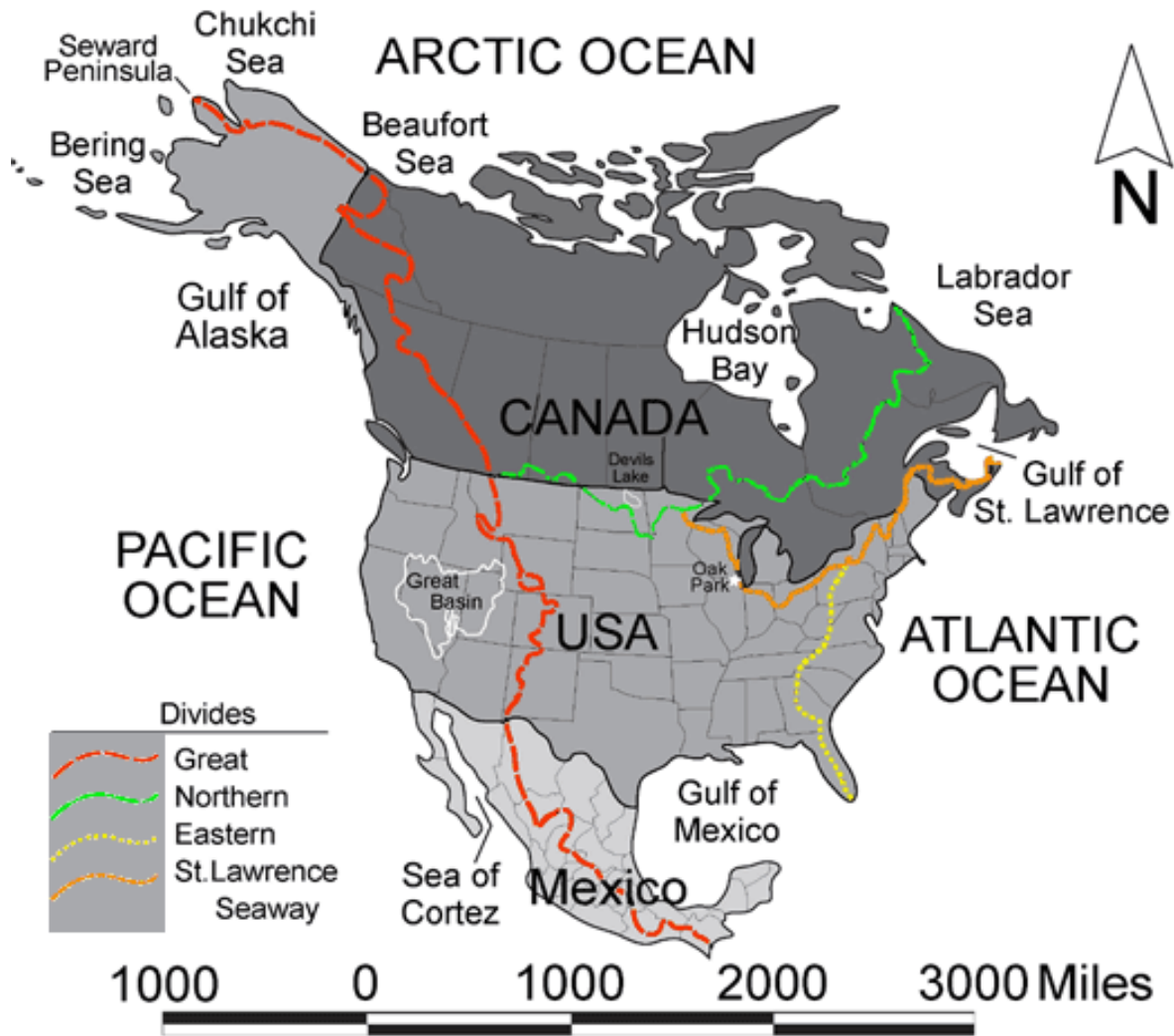


Polling Question #1

How did you hear about this webinar?

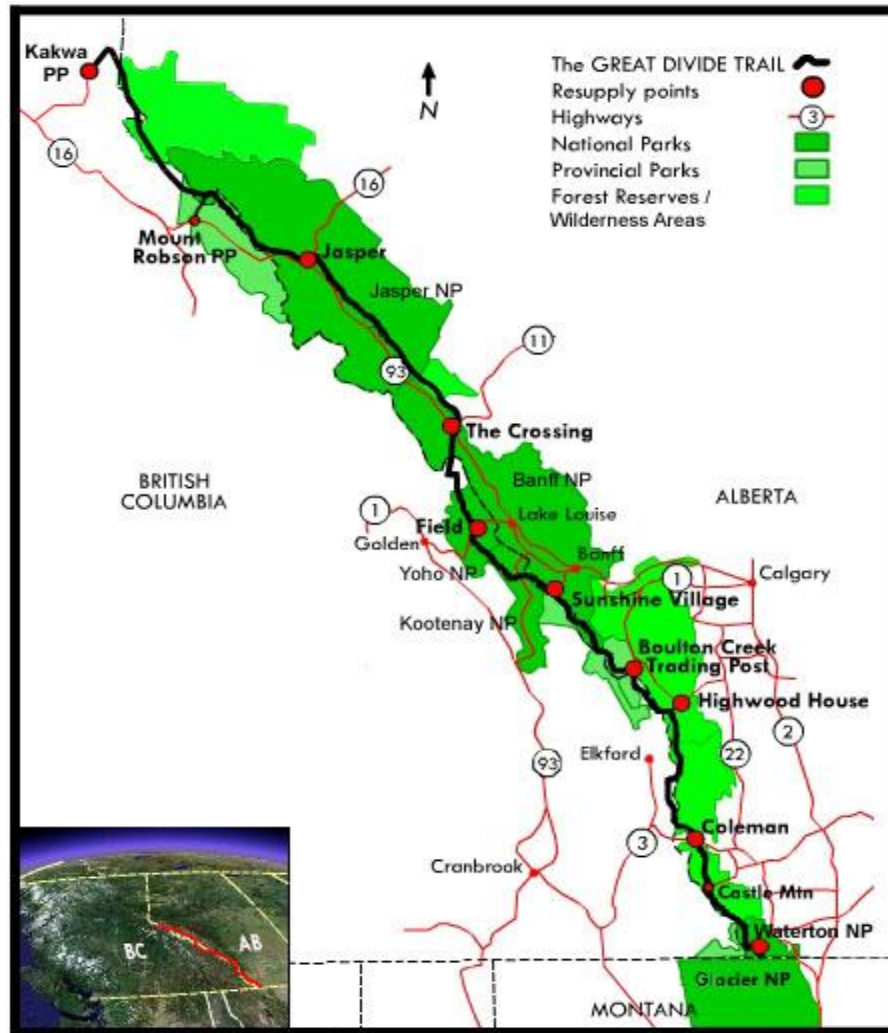
- 1.Social Media (i.e., LinkedIn, Twitter)
- 2.SEI Website
- 3.SEI Member Bulletin
- 4.Email invitation from the SEI
- 5.Website with webinar calendar (i.e., www.webinar-directory.com)





http://en.wikipedia.org/wiki/Continental_Divide_of_the_Americas





http://en.wikipedia.org/wiki/Great_Divide_Trail



“The **Great Divide Trail**, or GDT, is a wilderness hiking trail in the Canadian Rockies. The trail closely follows the Continental Divide between Alberta and British Columbia, crossing the divide no fewer than 30 times. (...)

The GDT is not officially recognized by Parks Canada and therefore is not signed and not always even an actual trail, sometimes merely a wilderness route.”*



* http://en.wikipedia.org/wiki/Great_Divide_Trail



Crossing the Great Divide



Polling Question #2

Are you currently using Agile Development practices within your organization?

1. Yes
2. No
3. Not sure



Polling Question #3

Do you think that Agile Development and Software Architecture are

1. In conflict
2. Complementary
3. I'm not sure

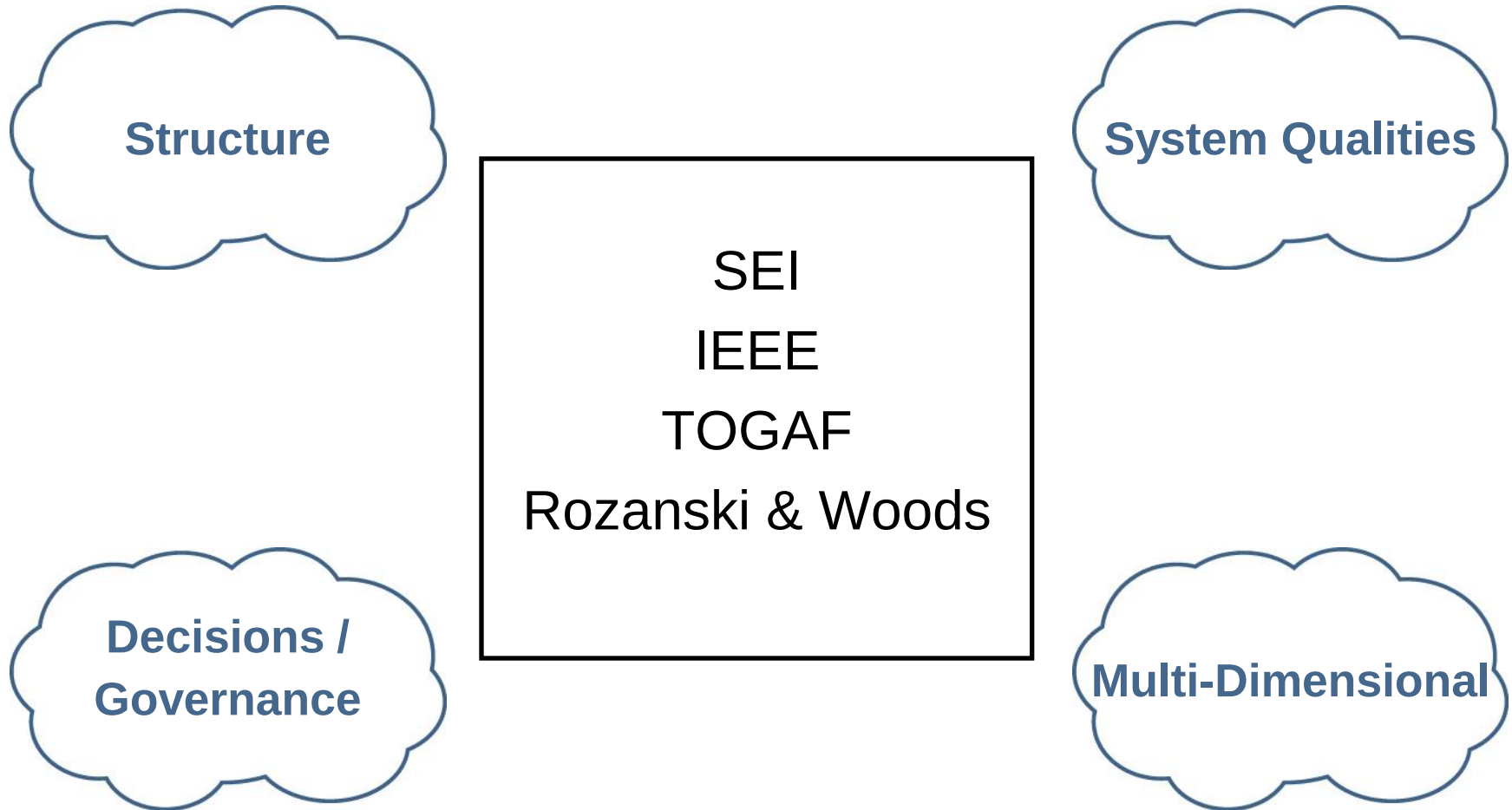


Scouting the Terrain

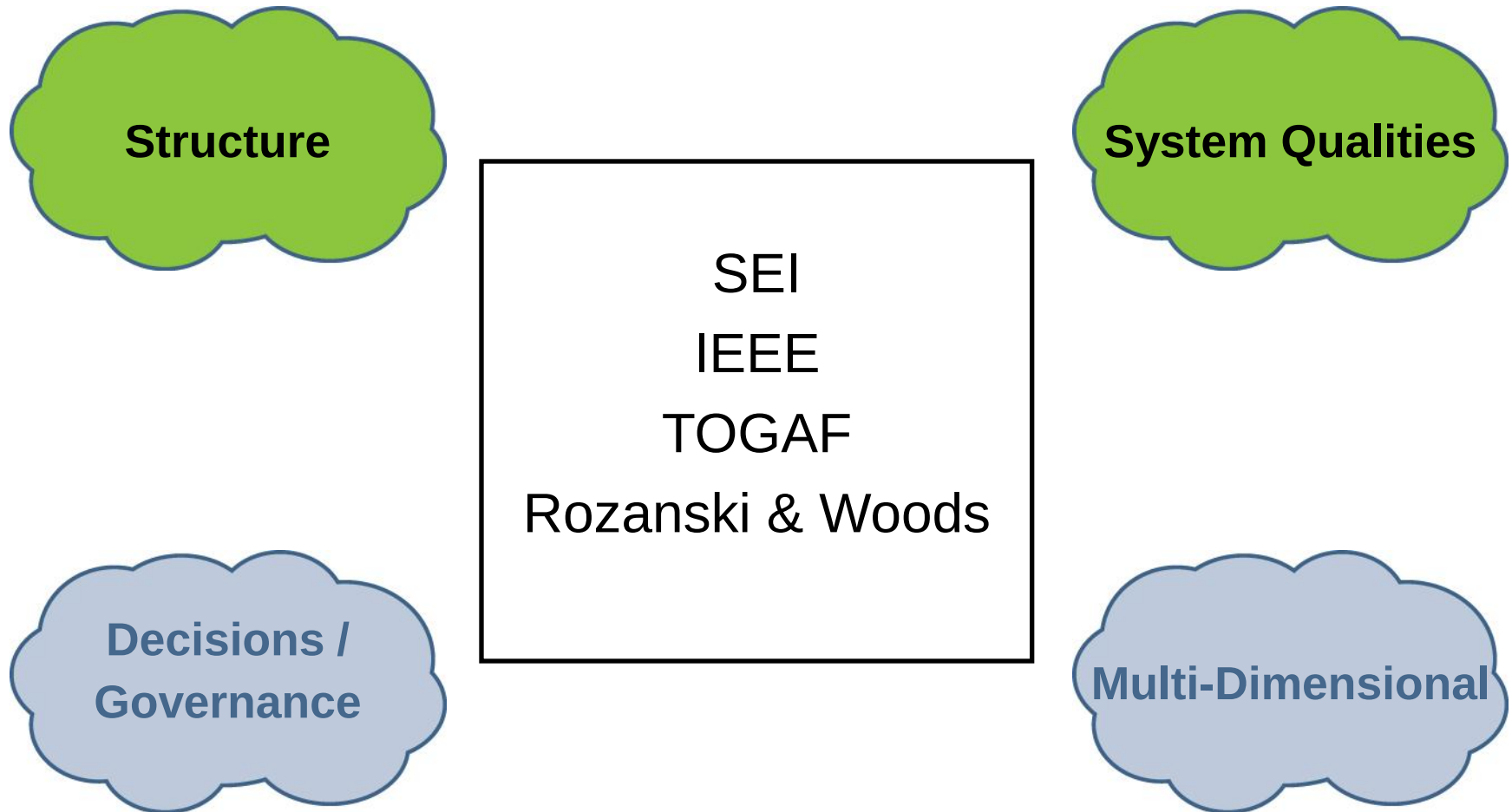


What is Architecture?

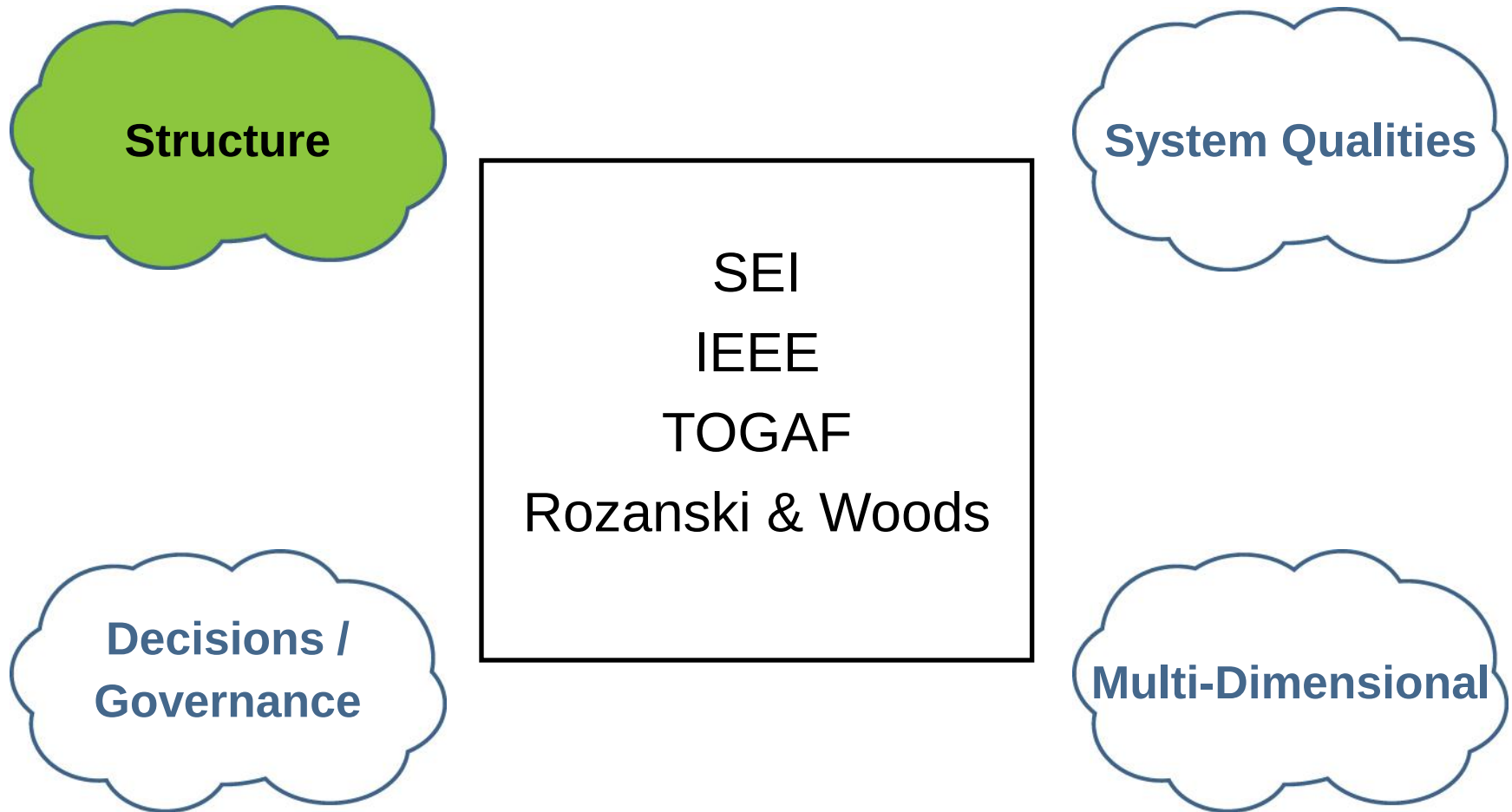
A Thematic Analysis



Architectural Themes



Architectural Themes



Structural Theme – Agreement

Both Agilists and Architects agree that that the structure of a system (i.e., the system's decomposition into components and component inter-relationships) is a real and significant concern.



Structural Theme – Debate

Debate between the Agile and Architectural communities focuses on the malleability of that structure and the extent to which it should be pre-defined or allowed to emerge throughout the course of development.



Structural Theme – Resolution

Context is key

- Project size – features, code, team size
- Product criticality



Alistair Cockburn

Crystal Family of Methodologies

Criticality	Life	L6	L20	L40	L80
	Essential Money	E6	E20	E40	E80
	Discretionary Money	D6	D20	D40	D80
	Comfort	C6	C20	C40	C80
		Team Size			



Structural Theme - Resolution

Context is key

- Organizational constraints
 - Geographical distribution
 - Culture – tolerance for ambiguity and risk, trust
- Discovery and innovation
 - New, unknown emerging market or well-established domain?
 - Maturity of technology / organizational technology experience base
- Technology
 - Embedded System vs. Enterprise Architecture
 - Flexible commercial product framework or “close to the metal” development environment

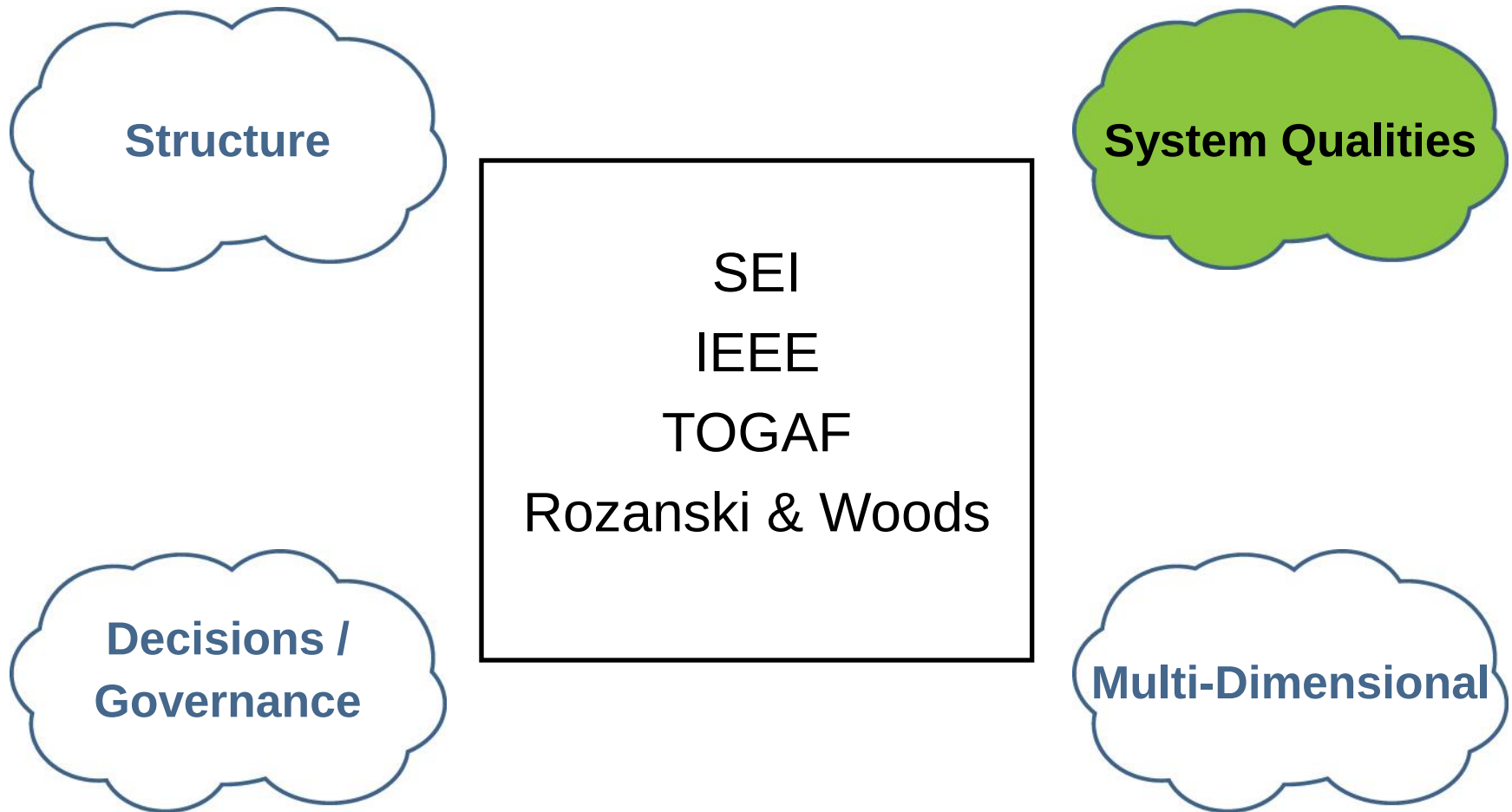


Philippe Kruchten Blog Post – “The Context of Software Development”

<http://pkkruchten.wordpress.com/2009/07/22/the-context-of-software-development/>



Architectural Themes



System Qualities Theme

Quality attributes are typically referred to as non-functional requirements. They represent system characteristics (i.e., qualities of a system) such as performance, availability, scalability, modifiability, and testability.



System Qualities Theme

Quality attributes may be classified by their primary stakeholders:

- End-user stakeholders
- Development stakeholders
- Delivery & support stakeholders



Examples of Quality Attributes

End-User Stakeholders

- Reliability
- Performance
- Availability
- Security
- Usability

Development Stakeholders

- Modifiability
- Testability
- Portability
- Reusability

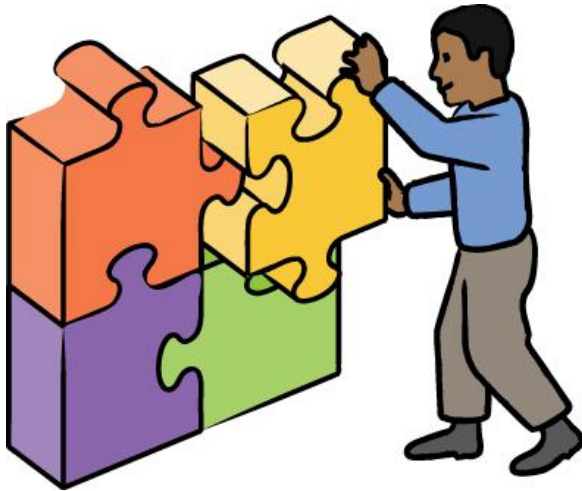
Delivery & Support Stakeholders

- Installability
- Diagnosability
- Per-Click-Sales-Ability
- SaaS-Ability
- Auditability
- Reconcilability



Eliciting / Expressing Quality Attributes

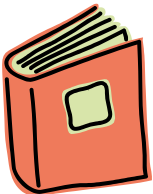
Quality Attribute Scenarios



Stimulus

Environment

Response



Bass, Clements, Kazman – “Software Architecture in Practice”



Eliciting / Expressing Quality Attributes



“Forgotten Stakeholder” Stories

As a <Stakeholder>, I want <some goal>
so that <some reason>”

Acceptance Tests

Given preconditions (Environment)

When actions or triggers (Stimulus)

Then consequences (Response)



Mike Cohn – “User Stories Applied”

Dan North – “Introducing BDD” - <http://blog.dannorth.net/introducing-bdd/>



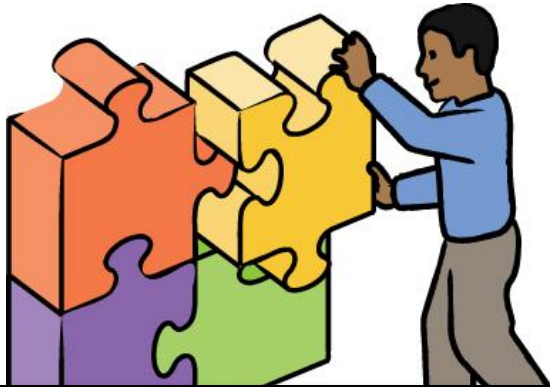
Polling Question #4

How are quality attributes (or non-functional requirements) elicited / expressed in your organization?

1. Declarative statements (the system shall ...)
2. Scenarios or stories
3. Test Cases
4. Not much attention explicitly paid to quality attributes
5. Other



When Should Quality Attributes be Addressed?



“The quality attributes of any nontrivial system are determined by its architecture.”¹



“Make it work, then make it faster.”²



1 Clements, Kazman, Klein – “Evaluating Software Architectures”, p. 109

2 Cohn – “Agile Estimating and Planning” p. 126 – reference to “The Elements of Programming Style” by Kernighan and Plauger



When Should Quality Attributes be Addressed?

Context is Key!



Tools for Navigation



Agile & Architecture in Release Planning

Release Planning is a critical life-cycle practice.

Release Planning forces choices that bring into focus issues of cost and value, current needs, and future potential.



Agile & Architecture in Release Planning

How to integrate architectural considerations into Agile release planning?

How to make the release planning for architecture more Agile?



Life Cycle and Release Planning

Life cycle has a major influence on the way in which architecture is addressed during release planning.



Architecture within the SDLC

Agile – Inter-weave architectural implementation with the implementation of stakeholder stories.

RUP – Design, code, and test architecture during Elaboration Phase iterations by focusing on architecturally significant requirements. Non-architecturally significant requirements are defined, implemented, and tested in the Construction Phase.

Waterfall – Define all requirements, complete all architecture and design, complete all coding, and perform all test activities.



Polling Question #5

Which of these descriptions most closely matches development practices at your organization?

1. Waterfall
2. RUP
3. Agile
4. None of the Above



Architecture in Agile Release Planning



Release Planning Inputs

Stakeholder Goals

- Include
 - Functionality
 - Quality attributes
 - Constraints
- Expressed by
 - End users
 - Development team
 - Delivery and support team
- Expressed as
 - Stories
 - Acceptance test cases



Release Planning Inputs

Technical Infrastructure

- Includes
 - Architectural implementation / enhancements
 - Technical research / technology selection
 - Code-level and architectural refactoring
 - Technical debt – incursion and reduction
- Expressed by
 - Development team
- Expressed as
 - Stories
 - Design spikes
 - Tasks
 - Technical models / sketches / documentation



Release Planning Inputs

Risks*

- Include
 - Technical risks
 - Market risks
 - End-user acceptance risks
 - Deployment risk, etc.
- Expressed by
 - End users
 - Development team
 - Delivery and support team
- Expressed as
 - Stories – “As a <Stakeholder> I want <risk mitigation action> so that <risk mitigation result>”

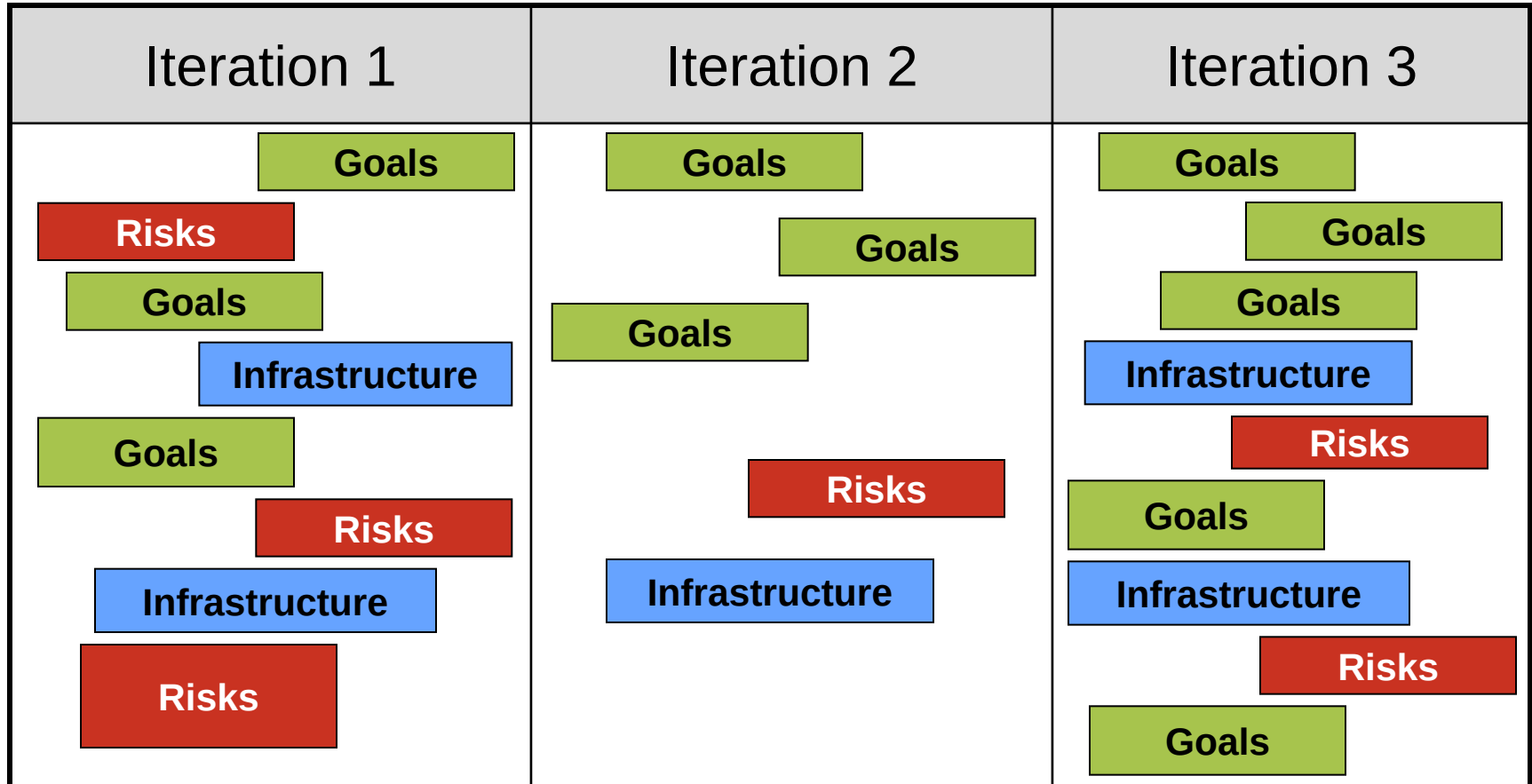
***Risks that influence release planning outcomes**



Architecture in Agile Release Planning



Release Planning Dashboard



Risk Tracking Sideboard

Identify	Mitigate	Retire
 		
	Monitor	React
		



The Question of Real Options

Should I make an architectural investment in anticipation of a future need for a given story? (*i.e., functional or quality attribute story*)

How likely is it that the future need will arise?

Is there an architectural investment that I can make now that will reduce the future cost to implement the story?

If so, what is the cost of this architectural investment?
(*e.g., cost to implement, opportunity cost, etc.*)

What are the relative economics of meeting the future need with or without having made the prior architectural investment?
(*e.g., relative cost and time to implement with or without prior architectural investment, potential opportunity cost from delay in meeting the future stakeholder need, etc.*)



Technical Debt

Technical Debt is a metaphor developed by Ward Cunningham as a means of explaining the need for refactoring to non-technical product stakeholders.



Cunningham, *The WyCash Portfolio Management System*. OOPSLA '92 Experience Report,
<http://c2.com/doc/oopsla92.html>



Technical Debt

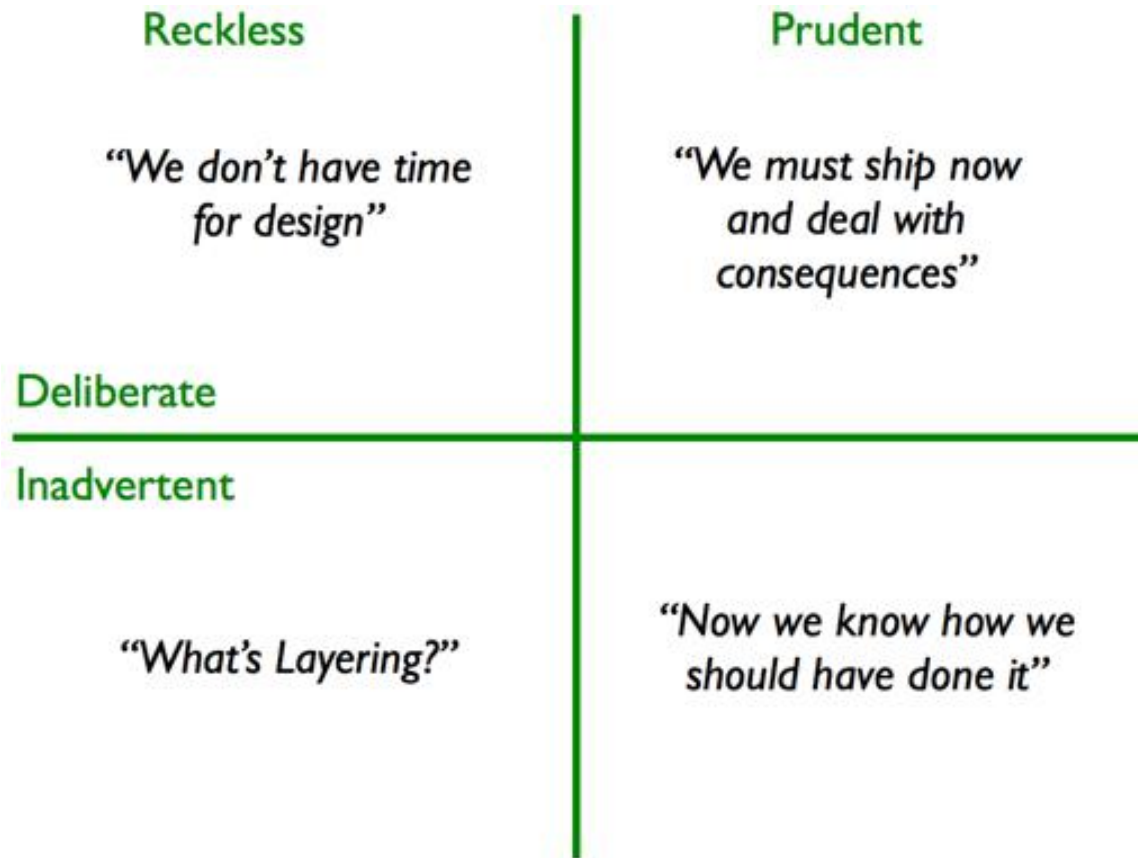
Releasing a system with suboptimal architecture, design and / or code burdens the development organization with debt.

The interest payments associated with the debt cause future system enhancements to require increased time and effort.

If re-factoring techniques are not used to pay down the debt, debt can continue to accumulate to the point where enhancement activities grind to a halt, resulting in metaphorical (and potentially literal) bankruptcy.



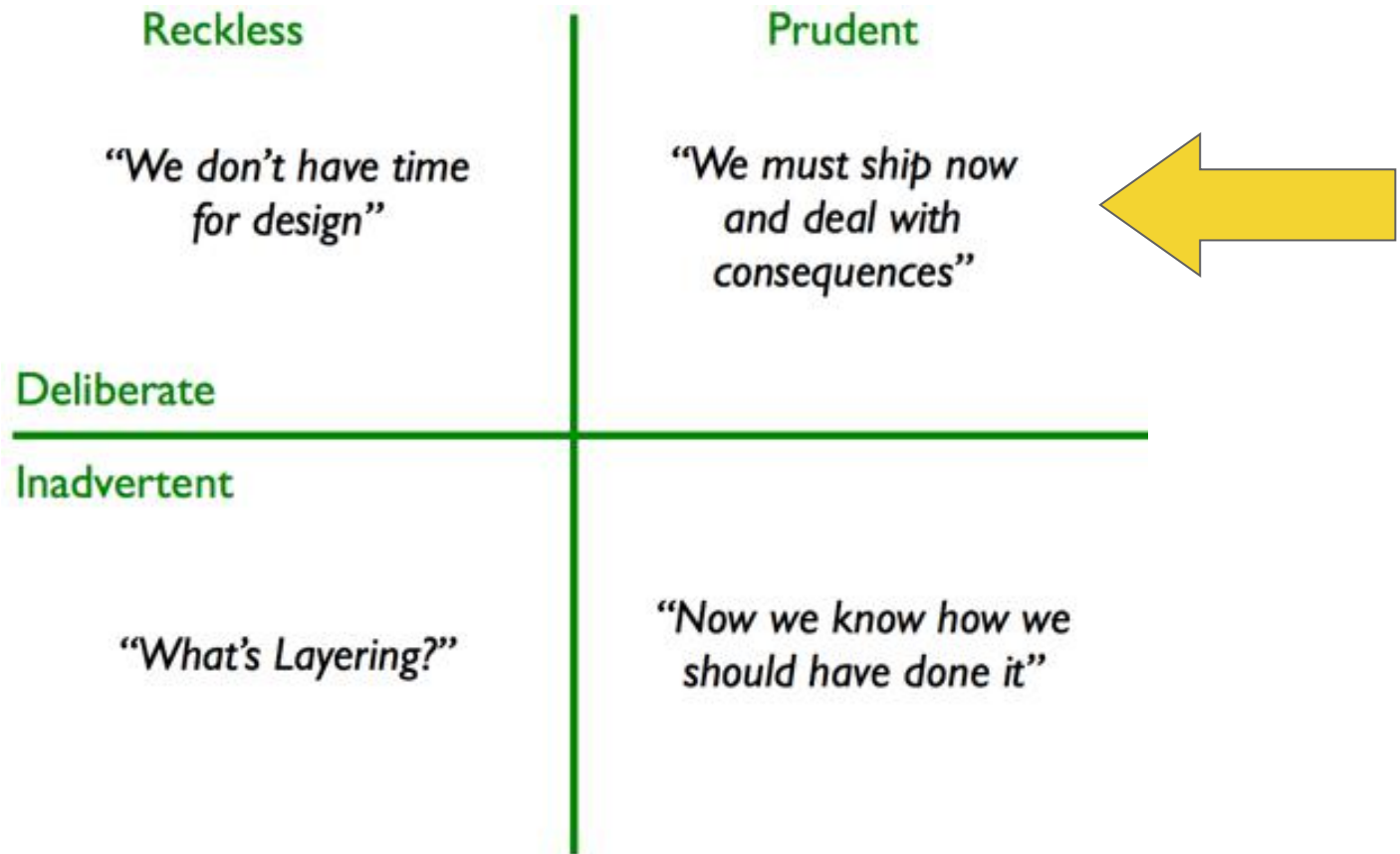
Martin Fowler's Taxonomy of Technical Debt



<http://martinfowler.com/bliki/TechnicalDebtQuadrant.html>



Real Options and Technical Debt



<http://martinfowler.com/bliki/TechnicalDebtQuadrant.html>



Considering Dependencies in Agile Release Planning

Dependencies between stories & supporting architectural elements

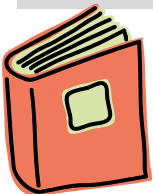
Understanding the dependencies between stories and architectural elements enables staged implementation of technical infrastructure in support of achieving stakeholder value.

Dependencies between architectural elements

Low-dependency architectures are a critical enabler for scaling-up agile development.¹

Dependencies between stories

High-value stories may require the implementation of lower-value stories as precursors.²



1 Mary and Tom Poppendieck – “Leading Lean Software Development”

2 Mark Denne, Jane Cleland-Huand – “Software by Numbers”



Where Do We Go From Here?

Focus on:

Design of agile practices for

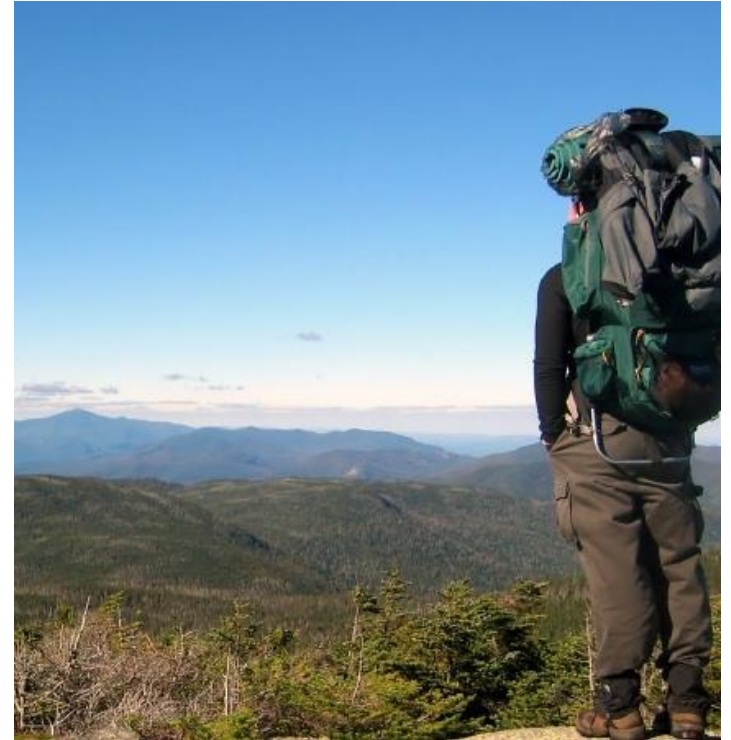
- Real options analysis
- Quantifying architectural value
- Dependency analysis
- Managing technical debt
 - Incursion (real options analysis)
 - Reduction (enhancement of technical infrastructure)



We Hope You Will Join Us On the Journey

We Welcome

- Comments
- Questions
- Critiques
- Ideas
- Anecdotes
- Experience Reports
- Collaboration Opportunities



<http://saturnnetwork.wordpress.com/>

Nanette Brown
nb@sei.cmu.edu



Discuss Agile Development and Software Architecture further at SATURN 2010

Keynotes and invited talks

Jim Highsmith

Architects: Accelerators or Anchors to Organizational Agility?

Wayne Longcore

*Managing scale and agility:
Transformational Architecture for the Smart Grid*

Philippe Kruchten

*Software architecture and agility:
a clash of two cultures?*



Discuss Agile Development and Software Architecture further at SATURN 2010

Presentations

Architecture and Agile, Friends or Enemies?

Ger Schoeber, Sioux Embedded Systems B.V.

Designing and Building Large-Scale Systems in an Agile World

Stevie Borne, Dave Henricksen, Thomson Reuters,

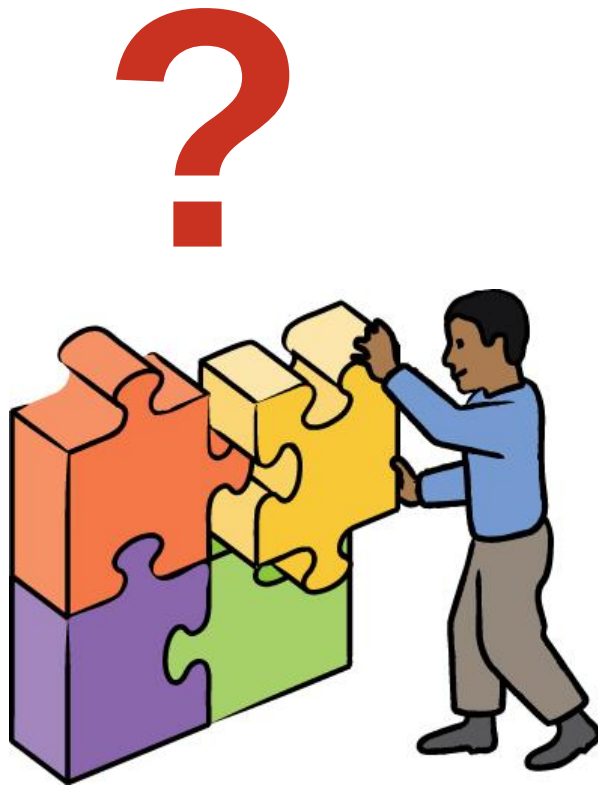
Agile Architecting: Using Agile Principles to Agilitize the Architecting Process

Amine Chigani, Virginia Tech

Agile Architect - Integrating Enterprise Architecture into Agile and Lean Software Development Environments

Srini Penchikala, InfoQ





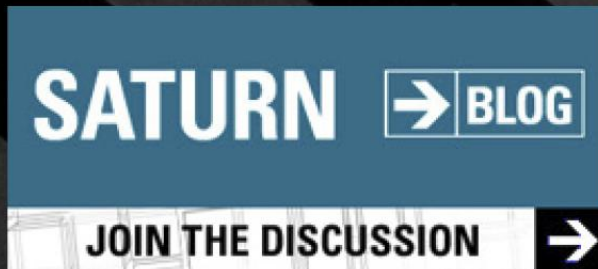
Questions??



Comments!!



JOIN THE CONVERSATION



The SEI Architecture Technology User Network (SATURN) is a professional network of software, systems, and enterprise architects from around the world, representing industry, academia, and government.



SATURN 2010

Sixth Annual SEI Architecture Technology
User Network Conference

ARCHITECTING FOR CHANGE

May 17-21, 2010
Minneapolis, Minnesota



REGISTER NOW



www.sei.cmu.edu/saturn/2010



Software Engineering Institute

Carnegie Mellon

TWITTER Hashtag #seiwebinar
Crossing the Great Divide
Brown, 4/22/2010

© 2010 Carnegie Mellon University



SEPGSM 2010
NORTH AMERICA

Perform at a Higher Level

SEPG Europe 2010

28 June - 1 July, 2010

Porto, Portugal



www.sei.cmu.edu/sepg/europe/2010/

SEPG is the premier, global conference series on software and systems process management



Software Engineering Institute

Carnegie Mellon

TWITTER Hashtag #seiwebinar
Crossing the Great Divide
Brown, 4/22/2010

© 2010 Carnegie Mellon University



Want a Closer Connection to the SEI?

Become an SEI Member!



www.sei.cmu.edu/membership



Software Engineering Institute

Carnegie Mellon

TWITTER Hashtag #seiwebinar
Crossing the Great Divide
Brown , 4/22/2010

© 2010 Carnegie Mellon University



Do you have the knowledge you need?

SEI Training



www.sei.cmu.edu/training



Software Engineering Institute

Carnegie Mellon

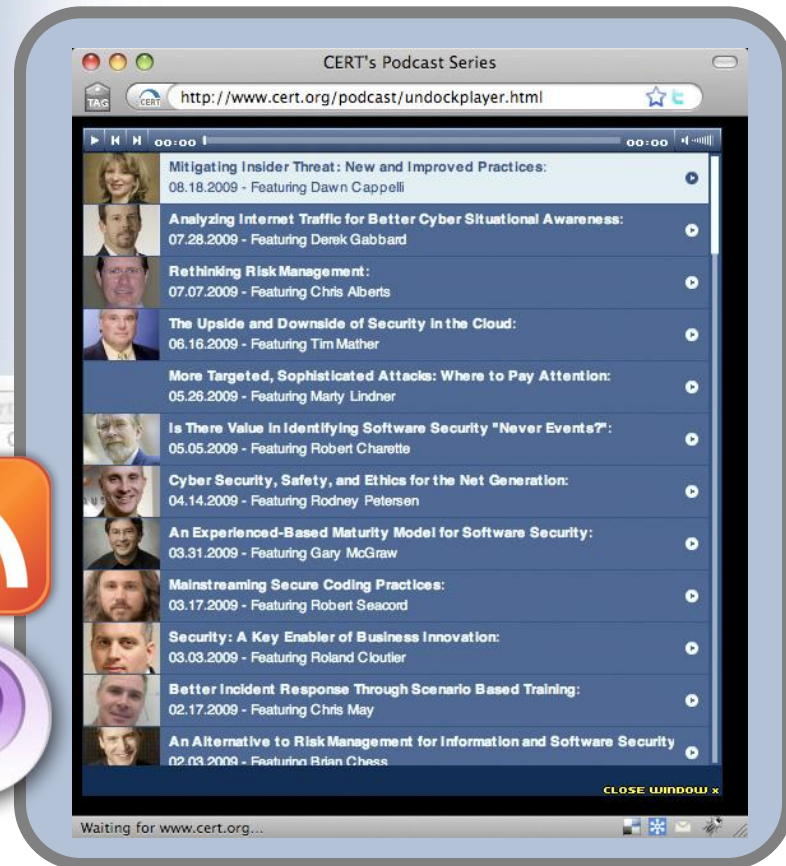
TWITTER Hashtag #seiwebinar
Crossing the Great Divide
Brown , 4/22/2010

© 2010 Carnegie Mellon University



CERT's Podcast Series: Security for Business Leaders

www.cert.org/podcast/

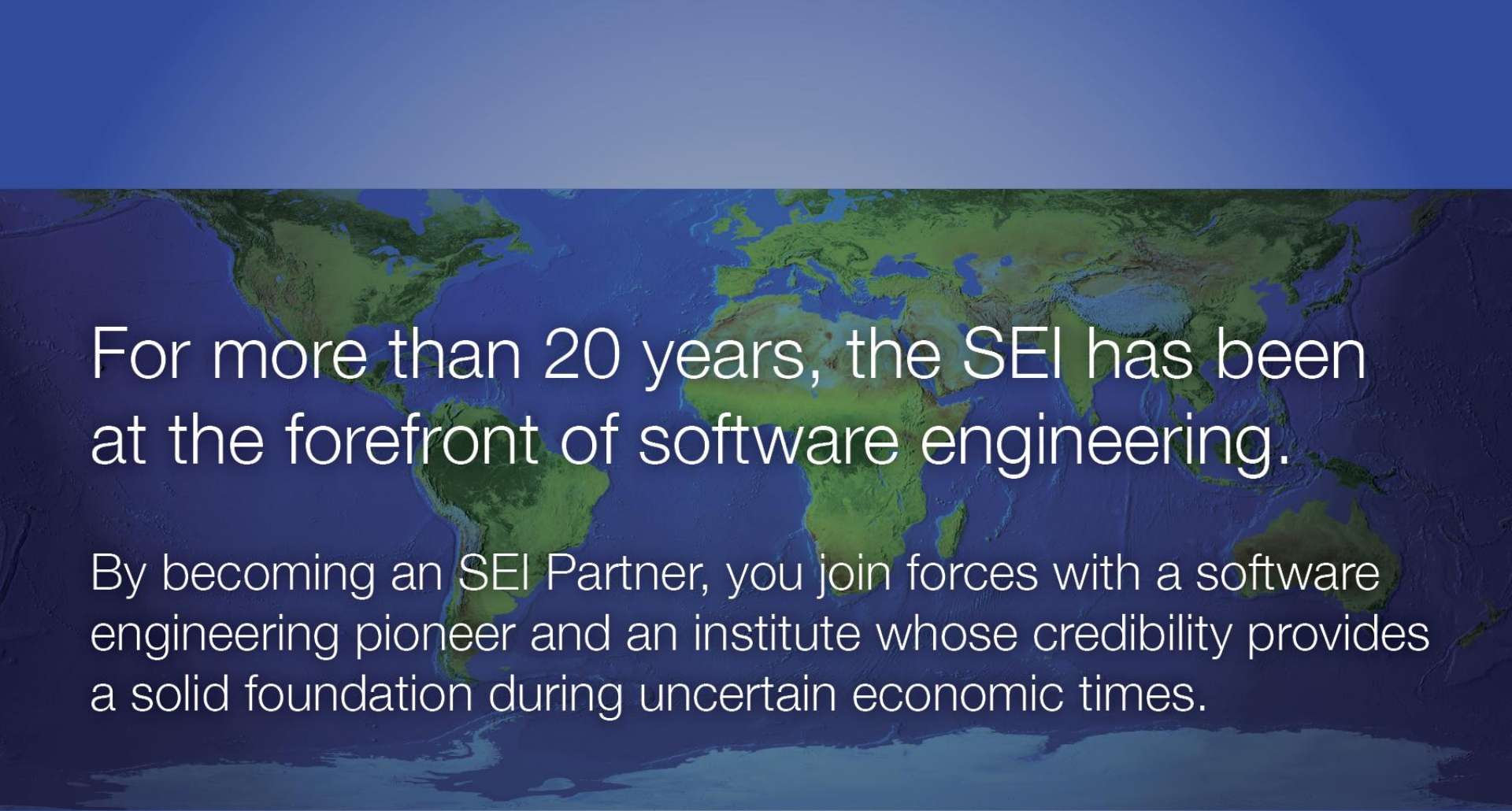


Software Engineering Institute

Carnegie Mellon

**TWITTER Hashtag #seiwebinar
Crossing the Great Divide
Brown, 4/22/2010**

© 2010 Carnegie Mellon University

A world map with a blue-to-green color gradient, showing the continents and oceans. The map is centered on the Atlantic Ocean, with North and South America on the left and Europe, Africa, and Asia on the right.

For more than 20 years, the SEI has been at the forefront of software engineering.

By becoming an SEI Partner, you join forces with a software engineering pioneer and an institute whose credibility provides a solid foundation during uncertain economic times.

SEI Partner Network

▶ www.sei.cmu.edu/partners



Software Engineering Institute

Carnegie Mellon

TWITTER Hashtag #seiwebinar
Crossing the Great Divide
Brown, 4/22/2010
© 2010 Carnegie Mellon University

ARCHITECTING FOR CHANGE

May 17-21, 2010
Minneapolis, Minnesota

Are you interested in learning more? Visit
<http://www.sei.cmu.edu/architecture/saturn/> to

SATURN  **about it** the SEI software architecture work, current research, tools and practices, news, and how the SEI can help you.

SATURN  **network** to architecture experts through the SATURN Network on LinkedIn.

• Attend SATURN 2010 – the annual conference that brings together experts from around the world to exchange best practices in developing, acquiring, and maintaining software, systems, and enterprise architecture. Registration is now open!



NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

